\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

76주차 - 데이터베이스 설계

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 데이터베이스 설계 시 고려사항

1) 무결성

- 연산 후에도 제약 조건 항상 만족

2) 일관성

- 특정 질의에 대한 응답이 처음부터 끝까지 변함없이 일정

3) 회복

- 장애 발생 직전 상태로 복구

4) 보안

- 불법적 데이터 노출 또는 손실로부터 보호

5) 효율성

- 응답시간 단축, 생산성, 최적화 가능

6) 데이터베이스 확장

- 운영에 영향주지 않으면서 지속적 데이터 추가 가능

2. 데이터베이스 설계 순서

1) 요구 조건 분석

- 데이터의 종류, 용도, 처리 형태, 제약 조건 수집

- 요구 조건 명세 작성

2) 개념적 설계(정보 모델링, 개념화)

- 추상적 개념으로 표현하는 과정

- 개념 스키마 모델링과 트랙잭션 모델링 병행 수행

- 요구조건 명세를 DBMS에 독립적인 E-R 다이어그램으로 작성

- DBMS에 독립적인 개념 스키마 설계

3) 논리적 설계(데이터 모델링)

- 컴퓨터가 이해할 있도록 논리적 자료 구조로 변환(mapping)

- DBMS에 따라 서로 다른 논리적 스키마 설계

- 트랜잭션 인터페이스 설계

- 관계형 데이터베이스라면 테이블을 설계하는 단계

4) 물리적 설계(데이터 구조화)

- 물리적 구조의 데이터로 변환

- 컴퓨터에 저장되는 방법 묘사

- 트랜잭션 처리량, 응답 시간, 디스크 용량, 저장공간 효율화 고려

5) 구현

- 데이터베이스 스키마를 파일로 생성하는 과정

- DDL을 이용해 스키마 기술 후 컴파일하여 빈 데이터베이스 파일 생성

- 데이터 입력

- 트랜잭션 작성

- 응용 프로그램 작성

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

77주차 - 데이터 모델의 개념

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 데이터 모델의 정의

1) 단순화, 추상화하여 체계쩍으로 표현한 개념적 모형

2) 데이터, 관계, 제약 조건 등을 기술하기 위한 개념적 도구들의 모임

3) 데이터 모델 구성 요소

- 개체, 속성, 관계

4) 데이터 모델 종류

- 개념,논리,물리적 데이터 모델

5) 데이터 모델에 표시할 요소

- 구조, 연산, 제약 조건

2. 데이터 모델의 구성 요소

1) 개체(Entity) : 현실 세계의 대상체

2) 속성(Attribute) : 데이터의 가장 작은 논리적 단위

3) 관계(Relationship) : 개체 간의 관계 또는 속성 간의 논리적 연결

3. 개념적 데이터 모델(정보 모델)

1) 추상적 개념으로 표현

2) 개체 타입과 관계를 표현

3) E-R 모델

4. 논리적 데이터 모델(데이터 모델)

1) 컴퓨터가 이해하고 처리할 수 있도록 변환

2) 데이터 간의 관계에 따라 관계 모델, 계층 모델, 네트워크 모델로 나뉨

3) 특정 DBMS는 특정 논리적 데이터 모델 하나만 사용

5. 논리적 데이터 모델의 품질 검증

1) 개체 품질 검증 항목

2) 속성 품질 검증 항목

3) 관계 품질 검증 항목

4) 식별자 품질 검증 항목

5) 전반적인 품질 검증 항목

6. 데이터 모델에 표시할 요소 ★★★★★

1) 구조 : 논리적으로 표현된 개체 타입들 간의 관계, 데이터 구조 및 정적 성질 표현

2) 연산 : 처리하는 작업, 데이터베이스 조작하는 기본 도구

3) 제약 조건 : 논리적인 제약 조건

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

82주차 - E-R(개체-관계) 모델

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. ER 모델의 개요

1) 개념적 데이터 모델, 피터 첸

2) 특정 DBMS를 고려한 것이 아님

2. ER다이어그램

1) 도형 외워라

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

84주차 - 관계형 데이터베이스의 구조

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 관계형 데이터베이스의 개요

1) IBM에 근무하던 코드(Codd)가 발명

2) 장점 : 보기 편하고 다른 데이터베이스로의 변환이 쉽다

단점 : 성능이 다소 떨어짐

2. 관계형 데이터베이스 Relation 구조

1) 튜플

- 각각의 행

- 속성의 모임으로 구성된다.

- 튜플의 수 = 카디널리티,기수,대응수

2) 속성

- 가장 작은 논리적 단위

- 속성의 수 = 디그리,차수

3) 릴레이션 스키마

- 속성 집합

4) 릴레이션 인스턴스

- 각 데이터 하나하나를 말함

5) 도메인

- 원자값들의 집합(범위)

3. 릴레이션 특징

- 릴레이션에 포함된 튜플들은 모두 상이하다

- 릴레이션은 시간에 따라 변한다.

- 순서는 중요하지 않다.

- 속성의 명칭은 유일, 값은 같을 수 있다.

- 속성의 값은 원자값만 저장

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

85주차 - 관계형 데이터베이스의 제약 조건 - 키

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 개체 무결성 제약과 참조 무결성 제약 등이 해당된다.

2. 키의 개념 및 종류

1) 후보키,기본키,대체키,슈퍼키,외래키

3. 후보키

1) 기본키가 될 수 있는 속성들

2) 유일성, 최소성 만족

4. 기본키

1) 후보키 중 특별히 선택한 주키

2) 널 값을 가질 수 없음

5. 대체키(=보조키)

1) 후보키에서 기본키를 제외한 키

6. 슈퍼키

1) 유일성은 만족하지만 최소성을 만족하지 않는 키

7. 외래키

1) 다른 릴레이션의 기본키를 참조

2) 기본키에 없는 값은 입력 불가

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

86주차 - 관계형 데이터베이스의 제약 조건 - 무결성

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 무결성(Integrity)의 개념 및 종류

1) 데이터 값과 실제 값이 일치하는 정확성

2) 부정확한 자료가 저장되는 것을 방지하기 위한 제약 조건

3) 종류에는 개체 무결성, 도메인 무결성, 참조 무결성, 사용자 정의 무결성

2. 개체 무결성(실체 무결성)

1) 기본키를 구성하는 어떤 속성도 널값이나 중복값을 가질 수 없다.

3. 도메인 무결성(영역 무결성)

1) 속성 값은 정의된 도메인에 속한 값이어야 한다.

4. 참조 무결성

1) 외래키 값은 널이거나 참조 릴레이션의 기본키 값과 동일해야 한다.

2) 참조할 수 없는 외래키 값을 가질 수 없다.

5. 사용자 정의 무결성

1) 속성 값들이 사용자가 정의한 제약 조건을 만족해야 한다.

6. 데이터 무결성 강화

1) 무결성은 데이터베이스 구축 과정에서 정의한다.

2) 애플리케이션, 데이터베이스 트리거, 제약 조건을 이용해 강화한다.

- 애플리케이션

- 데이터 조작 시 무결성 조건을 검증하는 코드 추가

- 애플리케이션 내에서 처리

- 장점 : 사용자 정의 같은 복잡한 무결성 조건 구현 가능

- 단점 : 소스 코드에 분산되어 있어 관리가 힘들고, 개별적인 시행으로 인해 적정성검토가 어려움

- 데이터베이스 트리거

- 트리거 이벤트에 무결성 조건을 실행하는 절차형SQL을 추가

- 장점 : 통합 관리가 가능하고, 복잡한 요구 조건의 구현 가능

- 단점 : 운영 중 변경이 어렵고, 주의가 필요

- 제약 조건

- 제약 조건을 설정하여 무결성 유지

- 장점 : 통합 관리 기능, 간단한 선언으로 구현 가능, 변경 쉬움, 오류 방지

- 단점 : 복잡한 조건 구현과 예외 처리 불가

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

87주차 - 관계대수 및 관계해석

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 관계대수의 개요

1) 정보를 검색하기 위해 어떻게 유도하는가를 기술하는 절차적인 언어

2) 연산의 순서를 명시

3) 관계 데이터베이스에 적용하기 위해 특별히 개발한 순수 관계 연산자와 수학적 집합 이론에서 사용하는 일반 관계 연산자가 있다.

- 순수 관계 연산자 : Select, Project, Join, Division

- 일반 관계 연산자 : UNION, INTERSECTION, DIFFERENCE, CARTESIAN PRODUCT

2. Select

1) 선택 조건을 만족하는 튜플의 부분집합을 구해 새 릴레이션 만드는 연산

2) 수평 연산

3) 시그마(σ) 사용

4) 표기 형식 σ<조건>(R)

- R은 릴레이션 이름

- 조건에는 비교 연산, 논리 연산 가능

- AND(∧), OR(∨), NOT(￢)

3. Project

1) 제시된 속성 값만을 추출

2) 수직 연산

3) 파이(π) 사용

4) 표기 형식 π<속성리스트>(R)

4. Join

1) 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합침

2) 새 릴레이션의 차수 = 두 릴레이션의 차수를 합침

3) join의 결과는 카디션 곱 수행 후 Select 수행한 것과 같음

4) (▷◁) 사용

5) 표기 형식 R▷◁(키속성r=키속성s)S

6) 자연 조인

- 같은 속성과 도메인을 가질 때 같은 속성이 한번만 나오게 새 릴레이션 구성

5. Division

1) 표기 형식 R[속성r ÷ 속성s]S

- R 릴레이션에서 속성 r을 빼고 S와 관련된 튜플만 보여줌

6. 일반 집합 연산자

1) 수학적 집합 이론

- UNION : 합집합

- INTERSECTION : 교집합

- DIFFERENCE : 차집합

- CARTESIAN PRODUCT : 그냥 곱

7. 관계 해석

1) 코드가 개발

2) 관계 데이터의 연산을 표현하는 방법, 비절차적 특성

3) 튜플 관계해석, 도메인 관계해석이 있다.

4) 관계대수로 표현한 식은 관계해석으로 표현할 수 있다.

5) 질의어로 표현

6) 주요 논리 기호

- ∀ : for all, 가능한 모든 튜플, 전칭 정량자

- ∃ : There Exists, 하나라도 일치하는 튜플이 있음, 존재 정량자

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

88주차 - 정규화

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 정규화의 개요

1) 관계형 스키마를 더 작은 속성의 세트로 쪼갠다.

2) 논리적 설계 단계에서 수행한다.

2. 정규화의 목적

1) 안정성 및 무결성

2) 효과적인 검색 알고리즘 생성

3) 데이터 중복 배제

3. 이상(Anomaly)의 개념 및 종류

1) 데이터들이 불필요하게 중복되어 릴레이션 조작 시 예기치 못한 현상 발생

- 삽입 이상 : 의도와는 상관없는 값이 함께 삽입

- 삭제 이상 : 의도와는 상관없이 연쇄적으로 삭제

- 갱신 이상 : 일부 튜플만 갱신되어 모순이 생김

4. 정규화 원칙

1) 정보의 무손실

2) 하나의 독립된 관계성은 하나의 독립된 릴레이션으로 분리

3) 데이터 중복성 감소

5. 정규화 과정

1) 1NF : 도메인이 원자값

2) 2NF : 완전 함수적 종속 만족, 부분 함수적 종속 제거

3) 3NF : 이행적 종속 제거, (A->B이고 B->C 일 때 C->A)

4) BCNF(Boyce-Codd 정규형) : 결정자이면서 후보키가 아닌 것 제거, 결정자가 모두 후보키인 상태

5) 4NF : 다치종속제거

6) 5NF : 조인 종속성

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

89주차 - 반정규화

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 반정규화의 개념

1) 시스템 성능, 편의성을 위해 의도적으로 정규화 원칙 위배하는 행위

2) 과도한 반정규화는 성능을 저하

3) 데이터의 일관성 및 정합성이 저하될 수 있다.

4) 방법에는 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가 등이 있다.

2. 테이블 통합

1) 조인되는 경우가 많아 합쳐 놓는 것

2) 1:1,1:N,슈퍼타입/서브타입 테이블 통합이 있다.

3) 테이블 통합 시 고려 사항

- 검색은 편하지만 레코드 증가로 처리량이 증가

- 데이터 조작이 복잡해질 수 있다.

- 제약조건 설계 어려움

3. 테이블 분할

1) 수직 또는 수평으로 분할

- 수직 분할

- 레코드 기준으로 분할

- 레코드별로 사용 빈도 차이가 큰 경우 빈도에 따라 분할

- 수평 분할

- 속성이 너무 많을 경우 분할

- 갱신 위주의 속성 분할 : 갱신이 자주 일어나는 속성 분할

- 자주 조회되는 속성 분할 : 자주 조회되는 속성이 극히 일부일 경우 분할

- 크기가 큰 속성 분할 : 이미지나 2GB 이상 속성들을 분할

- 보안을 적용해야 하는 속성 분할 : 보안 적용해야 하는 속성들 분할

2) 테이블 분할 시 고려 사항

- 기본키의 유일성 관리가 어려움

- 수행 속도가 느려질 수 있음

4. 중복 테이블 추가

1) 여러 테이블에서 데이터를 추출해서 사용하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우

2) 중복 테이블을 추가하는 경우

- 정규화로 인해 수행 속도가 느려지는 경우

- 많은 범위의 데이터를 자주 처리해야 하는 경우

- 특정 범위만 자주 처리해야하는 경우

- 범위를 줄이지 않고는 속도를 개선할 수 없는 경우

3) 중복 테이블 추가 방법

- 집계 테이블의 추가

- 테이블 생성하고 트리거 설정하는 것, 오버헤드에 유의

- 진행 테이블의 추가

- 이력 관리 등 적절한 데이터 양의 유지와 활용도 높이기 위해 기본키를 적절히 설정

- 특정 부분만을 포함하는 테이블의 추가

- 특정 부분만을 사용하는 경우 해당 부분만 새로운 테이블 생성

5. 중복 속성 추가

1) 조인해서 데이터 처리할 때 조회 경로 단축하기 위해 자주 사용하는 속성을 추가하는 것

2) 무결성 확보가 어렵고 디스크 공간이 추가로 필요함

3) 중복 속성을 추가하는 경우

- 조인 자주 발생, 접근 경로 복잡, 액세스 조건으로 자주 사용, 여러 개 속성으로 구성된 경우

4) 중복 속성 추가 시 고려 사항

- 테이블 중복과 속성의 중복 고려

- 일관성 및 무결성 유의

- SQL 그룹 함수를 이용해 처리

- 저장공간 지나친 낭비 고려

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

90주차 - 시스템 카탈로그

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 시스템 카탈로그의 의미

1) 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스

2) 시스템 카탈로그 내 테이블은 객체에 대한 정보를 유지 관리하는 시스템 테이블이다.

3) 카탈로그들이 생성되면 데이터 사전(Data Dictionary)에 저장됨

2. 시스템 카탈로그 저장 정보

1) 시스템 카탈로그에 저장된 정보를 메타 데이터라고 함

2) 메타 데이터 유형

- 데이터베이스 객체 정보 : 테이블, 인덱스, 뷰 등의 구조 및 통계 정보

- 사용자 정보 : 아이디, 비번, 접근 권한

- 테이블의 무결성 제약 조건 정보 : 기본키, 외래키

- 함수, 프로시저, 트리거 등에 대한 정보

3. 카탈로그 특징

1) 일반 이용자도 SQL 이용하여 내용 검색 가능

2) INSERT,DELETE,UPDATE로 카탈로그 갱신 불가

3) DBMS가 스스로 생성하고 유지

4) 카탈로그의 갱신 : 사용자가 SQL문을 실행시켜 기본 테이블, 뷰 인덱스 등에 변화를 주면 시스템이 자동으로 갱신

5) 분산 시스템에서의 카탈로그 : 위치 및 중복 투명성 제공하기 위해 모든 제어 정보를 가져야 한다.

4. 카탈로그/데이터 사전을 참조하기 위한 DBMS 내의 모듈 시스템

1) 데이터 정의어 번역기(DDL Compiler)

- DDL을 메타 데이터를 갖는 테이블로 변환하여 데이터 사전에 저장

2) 데이터 조작어 번역기(DML Compiler)

- 주 언어로 표현한 프로시저 호출로 변환하여 질의 처리기와 상호 통신

3) Data Dictionary

- 수록된 데이터를 실제로 접근하는 데 필요한 정보를 관리 유지하는 시스템

- 시스템만 접근 가능

4) 질의 최적화기

- 사용자 요구를 효율적인 형태롤 변환 후 질의 처리 방안 모색

5) 트랜잭션 처리기

- 데이터베이스 자원을 배타적으로 이용할 수 있도록 한다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

93주차 - 트랙잭션 분석/CRUD분석

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 트랙잭션 정의

1) 하나의 논리적 기능을 수행하기 위한 작업의 단위

2) 병행 제어 및 회복 작업 시 처리되는 작업의 논리적 단위로 사용

2. 트랜잭션 상태

1) 활동(Active)

- 트랜잭션 실행 중

2) 실패(Failed)

- 트랜잭션 실행 중 오류 발생하여 중단

3) 철회(Aborted)

- 트랜잭션이 비정상적으로 종료되어 Rollback 연산 수행

4) 부분 완료(Partially Committed)

- 트랜잭션이 모두 성공적으로 실행 후 Commit 연산 실행 되기 직전 상태

5) 완료(Committed)

- 트랜잭션이 모두 성공적으로 실행 후 Commit 연산 실행

3. 트랜잭션 특징(원일격지)

1) 원자성 : 모두 실행되거나 전혀 반영되지 않아야 함

2) 일관성 : 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 변환한다.

3) 격리성 : 둘 이상이 실행될 경우 하나가 완전히 끝날때까지 다른 트랜잭션 연산이 끼어들 수 없다.

4) 지속성 : 성공적으로 완료하면 고장나더라도 영구적으로 반영되어야 한다.

4. CRUD 분석

1) Create, Read, Update, Delete

5. CRUD 매트릭스

1) 행에는 프로세스를 열에는 테이블을 행과 열이 만나는 위치에는 변화를 표시

2) 우선순위 C>D>U>R

6. 트랜잭션 분석

1) CRUD 매트릭스를 기반으로 트랜잭션 양 분석

2) 업무 개발 담당자가 수행

7. 트랜잭션 분석서

1) CRUD 매트릭스를 이용해 작성

2) 구성요소

- 단위 프로세스

- CRUD 연산

- 테이블명,컬럼명

- 테이블 참조 횟수

- 트랜잭션 수

- 발생 주기

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

94주차 - 인덱스 설계

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 인덱스의 개념

1) 데이터 레코드를 빠르게 접근하기 위해 <키 값, 포인터> 쌍으로 구성되는 데이터 구조

2) 데이터가 저장된 물리적 구조와 밀접한 관계가 있다.

3) 액세스 빠르게 수행, 접근 방법 제공

4) 삽입, 삭제가 수시로 일어나면 인덱스 개수를 최소화하는 것이 효율적

5) DDL을 이용해 생성 변경 제거 할 수 있다

6) 인덱스가 없으면 TABLE SCAN 발생

7) 클러스터드 인덱스 : 실제 데이터가 순서대로 저장, 인덱스 검색하지 않아도 빠르게 찾음, 한개의 릴레이션에 인덱스 한개

넌클러스터드 인덱스 : 인덱스 키 값만 정렬, 검색 속도 떨어짐, 한개의 릴레이션에 인덱스 여러개

2. 트리 기반 인덱스

1) 인덱스 저장하는 블록이 트리 구조를 이루는 것

2) B 트리 인덱스

- 루트 노드에서 하위 노드로 키 값의 크기를 비교

- 모든 단말 노드는 같은 레벨에 있음

- 오름차순으로 저장됨

- 브랜치 블록과 리프 블록으로 구성됨

- 브랜치 블록 : 분기를 위한 목적, 다음 단계 포인터 가짐

- 리프 블록 : 인덱스를 구성하는 컬럼 데이터와 해당 데이터의 행 위치를 가리키는 레코드 식별자로 구성됨

3) B+트리 인덱스

- 단말 노드가 아닌 노드로 구성된 인덱스 세트, 단말 노드로만 구성된 순차 세트로 구분됨

- 인덱스 세트에는 키 값을 찾아갈 수 있는 경로만 주어짐

- 순차 세트에 있는 단말 노드가 데이터 주소 알려줌

- 단말 노드만을 이용한 순차 처리 가능

3. 비트맵 인덱스

1) 인덱스 컬럼의 데이터를 비트 값으로 변환하여 인덱스 키로 사용

2) 키 값을 포함하는 로우 주소 제공하는 것이 목적

3) 분포도가 좋은 컬럼에 적합, 성능 향상

4) 효율적 논리 연산, 저장 공간 작음, 압축 효율 좋음

5) 다중 조건을 만족하는 튜플의 개수 계산에 적합

4. 함수 기반 인덱스

1) 함수를 적용하여 산출된 값 사용, B+ 트리, 비트맵 사용

2) 함수 적용해야 하므로 부하 발생

5. 비트맵 조인 인덱스

1) 다수의 조인된 객체로 구성된 인덱스

2) 비트맵 인덱스와 물리적 구조가 동일

6. 도메인 인덱스(=확장형 인덱스)

1) 개발자가 필요한 인덱스를 직접 만들어 사용

7. 인덱스 설계

1) 분명하게 드러난 컬럼에 대해 기본적인 인덱스 먼저 지정 후 개발 단계에서 필요한 인덱스 설계를 반복 진행

2) 설계 순서

- 인덱스 대상 테이블이나 컬럼 선정

- 인덱스 효율성 검토하여 인덱스 최적화 수행

- 인덱스 정의서 작성

8. 인덱스 대상 테이블 선정 기준

1) MULTI BLOCK READ(한번에 읽을 수 있는 블록의 수) 수에 따라 판단

2) 랜덤 액세스가 빈번한 테이블

3) 특정 범위나 순서로 조회가 필요한 테이블

4) 다른 테이블과 순차적 조인이 발생되는 테이블

9. 인덱스 대상 컬럼 선정 기준

1) 인덱스 컬럼의 분포도 10~15% 이내

- 분포도 = (컬럼값의 평균 Row 수/테이블 총 Row 수)\*100

2) 수정이 빈번하지 않은 컬럼

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

95주차 - 뷰 설계

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 뷰의 개요

1) 허용된 자료만을 제한적으로 보여주는 가상 테이블

2) 저장장치 내에 물리적으로 존재하지 않지만 사용자에게는 있는 것처럼 간주됨

3) 조인문 사용 최소화

2. 뷰의 특징

1) 데이터의 논리적 독립성을 제공한다.

2) 뷰를 만들기 위해 유도된 기본 테이블을 정의 테이블이라 함

3) 기본 테이블의 기본키를 포함한 속성 집합으로 뷰를 구성해야만 삽입,삭제,갱신 연산 가능

4) 생성 : CREATE, 삭제 : DROP

3. 뷰 장단점

1) 장점

- 논리적 데이터 독립성 제공

2) 단점

- 독립적인 인덱스 가질 수 없음

- 뷰의 정의를 변경할 수 없음

- 뷰로 구성된 내용에 대한 연산에 제약이 따름

4. 뷰 설계 순서

1) 대상 테이블 선정

- 인라인 뷰 방식으로 접근되는 테이블

- 인라인 뷰 : FROM 절 안에 사용되는 서브쿼리

2) 대상 컬럼 선정

3) 정의서 작성

5. 뷰 설계 시 고려 사항

1) 구조가 단순화 될 수 있도록 반복적으로 조인 설정

2) 보안 유지

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

97주차 - 파티션 설계

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 파티션의 개요

1) 대용량의 테이블이나 인덱스를 작은 논리적 단위인 파티션으로 나누는 것

2) 파티션키 또는 인덱스키에 따라 물리적으로 별도의 공간에 데이터가 저장됨

3) 데이터 처리는 테이블 단위, 데이터 저장은 파티션별로 수행됨

2. 파티션의 장단점

1) 장점

- 데이터 접근 시 액세스 범위를 줄여 쿼리 성능 향상

- 파티션 단위로 입출력 분산 가능

2) 단점

- 세심한 관리 요구됨

- 테이블 조인 비용 증가

- 용량이 작은 테이블에 파티셔닝 하면 오히려 성능 저하

3. 파티션 종류

1) 범위 분할

- 지정한 열의 값을 기준으로 범위를 지정하여 분할

- 특정 파티션에 데이터가 몰릴 수 있음

2) 해시 분할

- 해시 함수를 적용한 결과 값에 따라 데이터 분할

- 특정 데이터가 어디있는지 판달할 수 없다.

- 데이터가 고른 테이블에 좋음

3) 조합 분할

- 범위 분할한 다음 해시 분할

- 범위 분할한 파티션이 너무 클 경우

4) 목록 분할

- 지정한 열 값에 목록을 만들어 이를 기준으로 분할

5) 라운드 로빈 분할

- 레코드를 균일하게 분배

- 순차적으로 분배, 기본키 필요없음

4. 파티션키 선정 시 고려 사항

1) 테이블 접근 유형에 따라 파티셔닝이 이뤄지도록 선정

2) 이력성 데이터는 파티션 생성주기와 소멸주기를 일치시켜야 한다.

3) 매일 생성되는 날짜 컬럼, 백업 기준 날짜 컬럼, 파티션 간 이동 없는 컬럼, 데이터 분포가 양호한 컬럼

5. 인덱스 파티션

1) 파티션된 테이블의 데이터를 관리하기 위해 인덱스를 나눈 것

2) Local Partitioned Index : 테이블 파티션과 인덱스 파티션이 1:1 대응, 비교적 데이터 관리가 쉬움

Global Partitioned Index : 테이블 파티션과 인덱스 파티션이 독립적

3) Prefixed Partitioned Index : 인덱스 파티션키와 인덱스 첫 번쨰 컬럼이 같다

Non-Prefixed Partitioned Index : 인덱스 파티션키와 인덱스 첫 번쨰 컬럼이 다르다

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

99주차 - 분산 데이터베이스 설계

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 분산 데이터베이스 정의

1) 논리적으로는 하나의 시스템, 물리적으로는 네트워크를 통해 연결됨

2. 분산 데이터베이스의 구성 요소

1) 분산 처리기

- 자체적으로 처리 능력 가짐, 지리적으로 분산되어 있는 컴퓨터 시스템

2) 분산 데이터베이스

- 지리적으로 분산되어 있는 데이터베이스, 해당 지역 특성에 맞게 데이터베이스 구성

3) 통신 네트워크

- 분산 처리기들을 통신망으로 연결

3. 분산 데이터베이스의 목표(위중병장)

1) 위치 투명성 : 실제 위치를 알 필요 없이 논리적 명칭만으로 액세스 가능

2) 중복 투명성 : 여러 곳에 중복되어 있어도 사용자는 하나만 있는 것처럼 사용, 시스템은 자동으로 여러 자료에 대해 수행

3) 병행 투명성 : 다수의 트랜잭션이 동시에 실행해도 트랜잭션 결과는 영향 받지 않는다.

4) 장애 투명성 : 장애에도 불구하고 트랜잭션을 정확히 처리

4. 분산 데이터베이스의 장단점

1) 장점

- 지역 자치성이 높음

- 점진적 시스템 용량 확장 쉬움

2) 단점

- 기능 복잡, 비용 증가

- 잠재적 오류 증가

5. 분산 데이터베이스 설계

1) 전역 관계망을 논리적 측면에서 소규모 단위로 분할, 그 결과를 복수의 노드에 할당

2) 노드에 할당된 소규모 단위를 분할(Fragment)이라 함

3) 설계 방법에는 분산, 분할, 할당이 있음

6. 테이블 위치 분산

1) 테이블을 각기 다른 서버에 분산시켜 배치

2) 구조 변경하지 않고 중복되지 않게 배치

3) 해당 테이블이 놓일 서버를 미리 설정해야 됨

7. 분할(Fragment)

1) 데이터를 분할하여 분산하는 것

2) 분할 규칙

- 완전성 : 전체 데이터를 대상으로 분할

- 재구성 : 연산을 활용해 본래 데이터로 재구성 가능해야 함

- 상호 중첩 배제 : 분할된 데이터는 서로 다른 분할의 항목에 속하지 않아야 함

8. 할당(Allocation)

1) 동일한 분할을 여러 개의 서버에 생성

2) 중복이 없는 할당, 중복이 있는 할당

- 비중복 할당 방식

- 최적 노드 선택, 데이터베이스의 단일 노드에서만 분할이 존재

- 비용 증가, 성능 저하 등의 문제가 생김

- 중복 할당 방식

- 동일한 테이블을 다른 서버에 복제, 부분 복제와 완전 복제가 있음

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

101주차 - 데이터베이스 보안/암호화

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 데이터베이스 보안의 개요

1) 권한이 없는 사용자가 액세스하는 것을 금지하는 것

2. 암호화(Encryption)

1) 평문을 암호화 하는 것

2) 암호화, 복호화

3. 개인키 암호 방식(Private Key Encryption) = 비밀키 암호 방식

1) 동일한 키로 암호화,복호화

2) 대칭 암호 방식, 단일키 암호화 기법이라고 함

3) 종류

- 전위 기법, 대체 기법, 대수 기법, 합성 기법(DES, LUSIFER)

- DES : 64BIT 평문 블록을 56BIT 16개 키를 이용해 16회 암호 계산 후 64BIT 암호문 만듦

4. 공개키 암호 방식(Public Key Encryption)

1) 서로 다른 키로 데이터 암호화, 복호화

2) RSA 방법 이 있다.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

102주차 - 데이터베이스 보안-접근통제

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 접근통제

1) 데이터가 저장된 객체와 이를 사용하려는 주체 사이의 정보 흐름을 제한하는 것

2) 접근통제 기술에는 임의 접근통제, 강제 접근통제, 역할기반 접근통제가 있다.

- 임의 접근통제(DAC : Discretionary Access Control)

- 사용자의 신원에 따라 접근 권한 부여

- 데이터 소유자가 지정, 제어

- 권한을 받고 다른 사용자에게 권한 넘길 수도 있음

- GRANT(수여), REVOKE(회수)

- 강제 접근통제(MAC : Mandatory Access Control)

- 주체와 객체의 등급 비교하여 접근 권한 부여

- 시스템이 권한 결정

- 자기보다 높은 등급 : 모든 것이 불가능

자기랑 같은 등급 : 읽기, 수정, 등록 가능

자기보다 낮은 등급 : 읽기만 가능

- 역할 기반 접근통제(RBAC : Role Based Access Control)

- 사용자의 역할에 따라 접근 권한 부여

- 위의 두개의 단점을 보완함, 다중프로그래밍 환경에 좋음

- 중앙 관리자가 역할마다 권한 부여

3) 강제 접근통제(MAC)의 보안 모델

- 벨 라파듈라 모델

- 군대 보안 모델

- 자기보다 높은 등급 : 작성만 가능

자기보다 낮은 등급 : 읽기만 가능

- 비바 무결성 모델

- 벨 라파듈라 보완한 수학적 모델, 무결성 보장하는 최초 모델

- 클락-윌슨 무결성 모델

- 무결성 중심 상업용 모델, 사용자가 직접 객체에 접근 불가, 프로그램에 의해 접근 가능

- 만리장성 모델

- 서로 이해 충돌 관계에 있는 객체 간 정보 접근 통제

4) 접근통제 3요소

- 접근통제 정책

- 접근통제 매커니즘

- 접근통제 보안모델

2. 접근통제 정책

1) 누가,언제,어디서,무엇을,어떻게에 대한 허용여부 정의하는 것

- 신분 기반 정책

- 신분에 근거하여 접근 제한

- IBP : 최소 권한 정책, 단일 주체에게 하나의 객체에 대한 허가 부여

- GBP : 복수 주체에 하나의 객체에 대한 허가 부여

- 규칙 기반 정책

- 주체가 갖는 권한에 근거하여 객체의 접근 제한

- MLP : 사용자 및 객체별로 지정된 기밀 분류에 따른 정책

- CBP : 집단별로 지정된 기밀 허가에 따른 정책

- 역할 기반 정책

- GBP의 변형된 정책, 맡은 역할에 근거하여 접근 제한

3. 접근통제 매커니즘

1) 접근통제 목록 : 객체 기준 특정 객체에 대해 어떤 주체가 어떤 행위를 할 수 있는지 기록한 목록

2) 능력 리스트 : 주체 기준 주체에게 허가된 자원 및 권한 기록 목록

3) 보안 등급 : 등급 기반으로 접근 승인 여부 결정

4) 패스워드 : 자신임을 증명할 때 사용

5) 암호화 : 암호화하는 것

4. 접근통제 보안모델

1) 기밀성 모델(인가된 사용자만 접근 가능)

- 군사적 목적으로 개발된 최초의 수학적 모델, 기밀성 보장이 최우선

- 제약조건

- 단순 보안 규칙 : 자신보다 높은 등급 읽을 수 없음

- 스타 보안 규칙 : 자신보다 낮은 등급 쓸 수 없음

- 강한 스타 보안 규칙 : 자신이랑 같은 등급만 읽고쓰기 가능

2) 무결성 모델(인가된 사용자만 수정 가능)

- 기밀성 모델에서 발생하는 불법적인 정보 변경 방지

- 일관성 유지

- 제약조건

- 단순 무결성 규칙 : 자신보다 낮은 등급 읽을 수 없음

- 스타 무결성 규칙 : 자신보다 높은 등급 쓸 수 없음

3) 접근통제 모델

- 접근 통제 매커니즘을 보안 모델로 발전 시킨 것

- 접근 통제 행렬

- 행은 주체, 열은 객체, 권한여부

5. 접근통제 조건

1) 값 종속 통제

- 객체에 저장된 값에 따라 다르게 접근통제 허용할 때 사용

2) 다중 사용자 통제

- 다수의 사용자가 동시에 접근할 때 사용

3) 컨텍스트 기반 통제

- 특정 시간, 주소 등에 근거하여 접근 제어, 다른 보안 정책과 결합하여 사용

6. 감사 추적

1) 모든 활동 기록하는 기능

2) 부적절한 데이터 조작 파악위해 사용

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

104주차 - 스토리지

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 스토리지의 개요

1) 대용량 데이터 저장하기 위해 서버와 저장장치 연결하는 기술

2) DAS, NAS, SAN

2. DAS(Direct Attached Storage)

1) 서버와 저장 장치를 전용 케이블로 직접 연결, 외장하드

2) 서버에서 저장 장치 관리

3. NAS(Network Attached Storage)

1) 서버와 저장장치를 네트워크를 통해 연결

2) Ethernet 스위치를 통해 다른 서버에서도 접근 가능

3) 접속 증가 시 성능 저하

4) 별도의 파일 관리 기능이 있는 NAS Storage가 / 내장된 저장장치를 직접 관리

4. SAN(Storage Area Network)

1) DAS, NAS 장점 혼합, 서버와 저장 장치를 연결하는 전용 네트워크 별도로 구성

2) 초기 설치 시 비용이 많이 듦

3) 광 채널 스위치를 이용해 네트워크 구성